(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 October 2003 (02.10.2003)

PCT

(10) International Publication Number
WO 03/081461 A1

(51) International Patent Classification[7]: G06F 17/30, H04L 29/12, 12/56

(21) International Application Number: PCT/FI02/00257

(22) International Filing Date: 26 March 2002 (26.03.2002)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **WIRAL LTD.** [FI/FI]; Westendintie 1, FIN-02160 Espoo (FI).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **KUMPULAINEN, Juha** [FI/FI]; Merenneidontie 22 B 4, FIN-02320 Espoo (FI).

(74) Agent: **PATENT AGENCY COMPATENT LTD.**; Hämeentie 29, 4th FI, FIN-00500 Helsinki (FI).

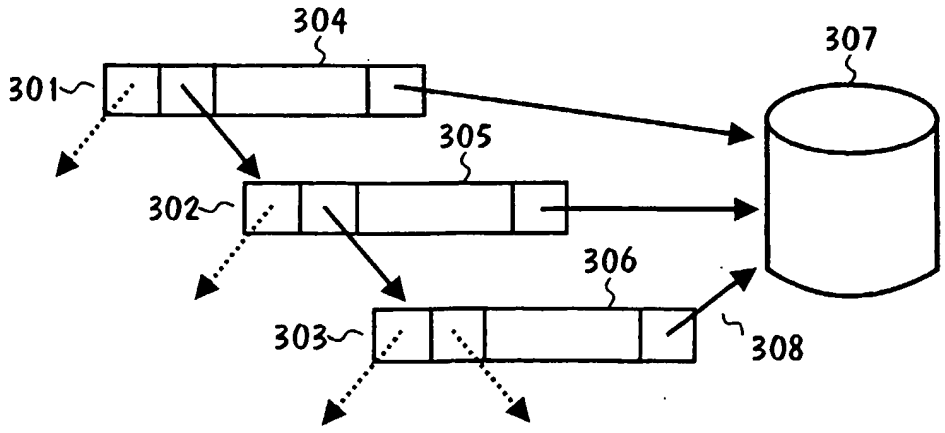
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SEARCH MEANS CONTAINING FIXED-LENGTH ADDRESSES GENERATED BY A HASH FUNCTION



(57) Abstract: Search means for database and file searches, wherein the search means combines the use of an appropriate data structure and the use of a hash function. The hash function generates fixed-length addresses which are located in the nodes of the data structure. The hash function/algorithm is preferably Digest and the data structure is, for example, a B-tree or skiplist. Another hash function and data structure can be used, too. Search means can be utilized in various database systems and especially in instant messaging services.

WO 03/081461 A1

Search means containing fixed-length addresses generated by a hash function

Field of the invention

The present invention relates to database searches and search means, especially used in an access point system composed of one or more servers, which are connected to each other by a intranet, and which handle data sent by client applications.

Background of the invention

Various client-server services are based on the use of access points, wherein clients are logged into a service and communicate with each other via a set of access points. Each client is an application program and a set of access points composes a logical server. Instant messaging services, such as a chat service and an e-mail service, are some examples of client-server services. A set of access points is connected to an access network that may be a fixed network or a wireless network. The access network is preferably a packet-switched network.

The Internet is a packet-switched network whose nodes have an Internet protocol address (IP address). Each IP address consists of four numbers between 1 and 255, and dots separating each number; for example, 193.199.35.5. The first number refers to the topmost network level and the second number refers to the next level, etc. The routers of the Internet locate the correct receiver by its IP address. Since the Internet is a packet-switched network, no circuit is allocated for the connection. Instead, data is transmitted in packets from the sender to the receiver.

Each IP packet includes a header with the following information: a packet length, time to live (hop counter), protocol, a sender IP address, a receiver IP address, a sender port (application), among other things. The IP packet may carry the packets of a higher level protocol as a payload. Typical higher level protocols are the transmission control protocol (TCP) and the user datagram protocol (UDP).

When using TCP/IP the bytes (octets) have a sequence number. Thus, a receiver node detects if one or more bytes are missing. Then it sends a retransmission request for missing bytes. In another protocol a receiver

may send an acknowledgement as response to receiving bytes. UDP is in some cases an alternative to TCP/IP. UDP implements means to send datagrams without any control protocol. Therefore retransmission requests or packet acknowledgements are not used in UDP. For the same reason a sender cannot know whether a receiver has received the packets sent.

The world wide web (WWW or web) is an Internet-based, distributed hypermedia information system. The web pages are traditionally represented using hypertext markup language (HTML). HTML and its successor, extensible markup language (XML), are intended for forming structured documents to be interchanged in the web. Structured documents are searched for and read through software that is termed a browser. Hypertext transfer protocol (HTTP) determines how structured documents are transferred in the Internet.

As the Internet has become very popular, it has also been brought to mobile and wireless devices. Many of the prior art services presently in use are based on the global system for mobile communications (GSM) standard. General packet radio services (GPRS) and the universal mobile telecommunications system (UMTS) are third-generation mobile communication systems which will replace second-generation mobile communication systems, such as GSM.

Wireless markup language (WML) is a formal language that allows the text portions of structured documents to be presented via a wireless network on wireless devices. WML is a part of wireless application protocol (WAP). WAP is similar to TCP/IP based protocols enabling Internet in wireless devices.

In addition to TCP/IP and UDP packets, short messages are a method to transmit data from a client to an access point. GSM limits the length of short messages to 160 characters. Multimedia messaging service (MMS) is able to deliver larger messages in a reasonable time compared to SMS. The fixed limit will be replaced by an ability to not only transfer much larger text contents, and graphics, but also audio or video clips.

Thus, the access network may be a fixed network or a wireless network, such as a GSM, GPRS, or UMTS. Or the access network may be a wireless local area network (WLAN). Clients send data via the access network to a set of access points. Data may be located, for example, in a TCP/IP packet, an UDP packet, or in a short message.

When a client is logged into a service and sends data to an access point, the data contains at least one reference address. The said reference address identifies a sender, a receiver, or a service. The reference address may be a fully qualified domain name (FQDN), or it may be an MSISDN number, i.e. a mobile subscriber integrated service digital network number.

Thus, data sent by a client may include one ore more reference addresses which are, for example, MSISDN numbers or fully qualified domain names. An e-mail address, such as Alfa@wiral.com, is one example of an FQDN, but there are also other types of domain names.

The Internet consists of thousands of domains. Each domain has a domain name which is mapped to a certain IP-address. Several domain names may be mapped to the same IP-address. For example, domain names www.jypoly.fi and jkolamk.jkol.jypoly.fi are mapped to IP-address 193.199.35.1. Conversely, fully qualified domain names are unique, such as e-mail user names.

Domain names compose a hierarchical domain name system (DNS). The root of a DNS tree is nameless. Top-level domains are under the root: the original three-letter domains are .com, .net, .org, .edu, .int, .mil and .gov, plus two-letter top-level domains for each country. Under the top-level domains there are lower domains connected to the Internet. The Internet includes domain name servers mapping domain names to IP addresses.

Uniform resource locator (URL) is a system uniquely identifying each resource in the Internet, i.e. where each document or file is located.

A URL address consists of a domain name and a search path. For example, URL address "www.jypoly.fi/internet/jamk.nsf" consists of domain name "www.jypoly.fi" and search path "/internet/jamk.nsf".

A URL request consists of a protocol part and a URL address. For example, in the following URL request the protocol part is "http://" and the URL address is the before-mentioned "www.jypoly.fi/internet/jamk.nsf":

http://www.jypoly.fi/internet/jamk.nsf

A uniform resource identifier (URI) is an access path to a certain piece of information. A URI always contains a URL. For example, a URI may be an access path to a WWW page:

http://www.jypoly.fi/internet/jamk.nsf/www/779EBAC7D6A67A14C22567E7002B127A?OpenDocument

FIG. 1 shows a client-server system composed of one access point, wherein client Alfa 11 and client Beta 12 are logged into a service. Client Alfa and Beta are applications having their own reference addresses, such as e-mail addresses. During the use of the service client Alfa sends data via an access network 13 to client Beta's reference address. An access point 14 receives the packets and transmits them to client Beta. In FIG. 1 client Alfa is located in a mobile phone and client Beta is located in a laptop. In addition to these devices, a client could be located in, for example, a personal digital assistant (PDA), a personal computer, or a network server.

For example, Jabber IM server described in http://www.jabber.org can be used as an access point for instant messaging (IM) services. However, when a system should have high capacity, a set of access points is needed to handle data sent by clients. The access points can be coupled/connected to each other by means of an intranet.

The invention relates to a system composed of at least one server which could be e.g. a database server, or an access point. A system may include several access points when access points are coupled, directly or indirectly to each other, for example, by using an intranet of 100 Mbps Ethernet. Also a relative high capacity network may be blocked because of high load. Blocking of a network is one drawback of the prior art. Patent application PCTxxxxxxxxxx contains solutions for the said drawback.

Another drawback of the prior art concerns search means that are used in a server for database or file searches. The search means are the subject of this patent application.

A database is a collection of data organized in a fashion that facilities updating, retrieving, and managing the data. The data may consist of anything, including, but not limited to reference addresses. Various search means can be used for database retrievals. Lists, tree data structures, and hashing methods are typical examples of search means.

A key is a value intended for searching for a certain data collection in a database. Typically a big database cannot be entirely located in a main memory. Various data structures have been designed and implemented to index data collections of a database. If the content of a database is changeless, the fastest search means/method is binary search. When a database includes n keys the processing time of the search operation is $O(\log n)$.

A binary tree is a data structure based on binary search. If keys are short and of a fixed-length, such as social security numbers, the keys are preferably located in the nodes of a binary tree. However, if a database is big and keys are long the keys must be located in disk memory, which makes database searches very slow because each comparison operation causes a disk access.

FIG. 2 illustrates using a binary tree as a search means. As shown in FIG. 2, each node of a binary tree includes, a left and right link to other nodes, and a link to a key record. For example, node 201 includes a left link 202 to node 203, a right link 204 to node 205, and a link 206 to a key record 207. The key record 207 includes a key 208 and a link 209 to a data collection 210. In FIG. 2 a search key 210 is "Mike" and the key 208 related to the root node 201 of the binary tree is "Lisa". First the link 206 is followed to obtain the key 208 and then the keys 210 and 208 are compared. Because Mike alphabetically succeeds Lisa, the right link 204 of the node is followed to obtain the next node 205. The node 205 has a link to a key record 212 including the key "Mike", i.e. in this case the search key is found from the binary tree. We may suppose that the binary tree and key records can be kept in main memory. Then a comparison operation does not cause a disk access. Still following links and comparing various-length keys essentially increases the processing time of the search operation.

Hashing is another search means. It is based on the use of a hash function that inputs a character string related to a data collection and outputs a numeric value. If the character string is a unique value, the numeric value should be, too. The numeric value determines a bucket in which the said data collection is stored. Each bucket is composed of a fixed number of cells so that each cell includes one data collection. Buckets are usually stored in a disk memory. An adequate hash function spreads data collections uniformly into buckets. In that case a search operation requires one disk access in which a certain bucket is read in main memory. However, hashing may fail so that a lot of buckets are empty and simultaneously some buckets are overflowed. Overflow means that all the cells of a bucket are in use. In that case the bucket is usually chained to one or more data collections located in an overflow area. This increases the number of disk accesses and the processing time of a search operation.

Especially an FQDN type of reference address, which includes an URL, may be very long. The handling of long reference address takes more time than handling of shorter reference addresses, of course. Anyway, various-length reference addresses slow down database and file searches.

Summary of the invention

The objective of the invention is to upgrade the search means of a file or a database when keys are of various-length and possibly long. The keys may be reference addresses, such as fully qualified domain names.

The objective is reached by combining the use of a hash function with an appropriate data structure. The hash function is used for generating fixed-length addresses, which are located in the nodes of the data structure. The hash function is preferably Digest and the data structure is, for example, a B-tree or skiplist. Another hash function and other data structures can be used, too.

The hash function should input a various-length reference address and it should output a relatively short fixed-length address. In addition, the fixed-length address should be unique with high probability. Because the fixed-length addresses generated are short, they can be located in the nodes of a B-tree and the size of the B-tree is still so small that the B-tree can be kept in the main memory.

The inventive search means can be utilized in various database systems and especially in access point systems.

Brief description of the drawings

The invention is described more closely with reference to the accompanying drawings, in which

Figure 1 shows an access point communicating with clients,
Figure 2 illustrates using a binary tree as a search means,
Figure 3 shows a binary tree whose nodes include a fixed-length address,
Figure 4 shows a B-tree including fixed-length addresses,
Figure 5 depicts a hash function and its input and output,
Figure 6 shows an example of an access point system,
Figure 7A shows an example of a database system,
Figure 7B shows a search means for uniform resources identifiers.

Detailed description of the invention

Usually, a data collection is considered to a have just one unique piece of information which is termed a primary key. The data collection may include another unique piece of information that can be used as a secondary key. There may be several secondary keys so that a primary key and each secondary key are related to a certain search means. Search means may differ from each other. We have used the term "reference address" along with the term "key". A reference address may or may not be unique.

The records of a search means may or may not be termed nodes and those nodes may or may not compose a data structure which is termed a tree or a list in the prior art.

FIG. 3 shows a binary tree each of whose nodes includes a fixed-length address that is generated from a reference address by using a hash function. The binary tree is balanced as the binary tree shown in FIG. 2 and both the trees include the same number of nodes. In each node 301, 302, and 303 is located a fixed-length address, and these fixed-length addresses are marked with 304, 305, and 306. The fixed-length address 304 is generated from the reference address "Lisa" and the fixed-length address 305 is generated from the reference address "Mike". When the search key is "Mike", the search key is generated by the same hash function that has been used to generate fixed-length addresses 304, 305, 306, and the other fixed-length addresses are located in the binary tree. The binary tree is used as a search means as follows. First the search address generated is compared to the fixed-length address located in the node 301. The search address is the bigger one, thus the right link of the node 301 is used to obtain the node 302. Then the search address is compared to a fixed-length address located in the node 302. Now the search address and fixed-length address match, thus a data collection related to the search key "Mike" is obtained from the database 307 by using the link 308 of the node 302.

Digest hash function, a skiplist, and a B-tree are all known in the prior art. Digest hash function, or in more specific, the MD5 message-Digest algorithm is described in RFC1321 published by the Internet engineering task force (IETF). Digest hash function may result in a non-unique fixed-length address, though its input, i.e. a reference address, would be unique. However, the probability that Digest hash function results in a non-unique fixed-

length address is very small when reference addresses are unique. The said probability is only $(1/16)^{32}$.

If required, a generated fixed-length address can be searched in a data structure where active fixed-length addresses are stored in. If the said address is found in the data structure, a new fixed-length address is generated until the address is not found in the data structure. This way it is possible to ensure that all active fixed-length addresses are unique.

The structure and use of a skiplist is described in Communications of the ACM, 33(6):668-676, June 1990.

A B-tree belongs to a set of data structures termed "balanced trees". A red-black tree and an AVL-tree are other examples of balanced trees. A binary tree and balanced trees are basic data structures that are taught in the literature of computer science.

FIG. 4 shows a B-tree including fixed-length addresses. Numbers from 2 to 37 represent 16 bytes long fixed-length addresses. In this example, the B-tree is composed of five nodes. The nodes of the B-tree do not need to contain the same number of fixed-length addresses, more generally, the nodes of the B-tree do not need to contain the same number of nodes. As in FIG. 3, in addition to a fixed-length address, a node of the B-tree typically includes a link to a certain data collection stored in a database. These links and the database are omitted from FIG. 4.

In addition to the data structures shown in FIG. 3 and 4, there are a number of data structures where operation as a search means can be upgraded by placing fixed-length addresses in the nodes of the data structures.

FIG. 5 depicts a hash function and its input and output. The hash function 52 reads a relative long reference address 51 as a parameter and outputs a relative short fixed-length address 53. A reference address may be 1-500 bytes long, when a fixed-length address is preferably 16 bytes long. The hash function is preferably Digest, but another appropriate hash function can be used. The hash function should output a fixed-length address that is unique with very high probability.

An inventive data structure can be implemented as follows. A set of data collections is stored in a database or file, wherein each data collection contains at least one reference address. The data collections are passed through one by one as follows: 1) the reference address is obtained from a

data collection, 2) the fixed-length address is generated by applying a hash function to the reference address, 3) the fixed-length address is located in a node of the data structure, and 4) the pointing means of the node are set to point to the data collection containing the reference address from which the fixed-length address located in the node was generated. The pointing means may be e.g. as a pointer, link or index. The node may also include at least another pointing means. The said pointing means can be set to point to a certain node of the data structure depending on the category of the data structure and the values of the fixed-length address already contained in the data structure. The data structure category may be e.g. a B-tree or skiplist.

The invention is not limited to any specific type of databases or files, i.e. it is very general-purpose. The search means is especially useful in instant messaging, thus instant messaging is next discussed in more detail.

In an instant messaging service data sent by a client contains at least one reference address. The reference address could be e.g. the receiver's e-mail address, such as beta@wiral.com. The reference address is inputted as a parameter to a hash function which generates a 16 bytes long fixed-length address. The fixed-length address replaces the reference address in the data part of the IP-packet that is transmitted in the intranet. The intranet is preferably an Ethernet network and the packets to be transmitted in the intranet are preferably IP packets.

FIG. 6 shows a system that contains three access points, a load balancer, and a gateway. Clients Alfa 61 and Beta 62 can communicate with each other through the said system. The clients may send data via an access network 63 to a load balancer 64. The load balancer takes care that the three access points, 65, 66, and 67 are uniformly loaded. The three access points and a gateway 68 are connected to an intranet 69. When client Alfa logs into the system, we may suppose that the access point 65 generates a fixed-length address for client Alfa and locates the fixed-length address in a node. The node is added to a search means used by the access point 65. Correspondingly, when client Beta logs into the system, the access point 67 generates a fixed-length address for client Beta and locates the fixed-length address in a node. The node is added to another search means used by the access point 67. Client Alfa sends data containing a reference address, wherein the data could be e.g. an instant message. The access point 65 obtains the reference address from the data and generates a fixed-length

address. Then the access point 65 locates the fixed-length address and the payload in an IP-packet and sends the IP-packet to the intranet 69 using one-to-many transmission method, i.e. broadcast, multicast or anycast methods. All access points receive the IP-packet, obtain the fixed-length address from it, and searches the fixed-length address from the memory by using the search means. If the said address is found, i.e. the fixed-length address is the same as the access point 67 generated for client Beta, the data sent by client Alfa is transmitted to client Beta. In this use case, clients Alfa and Beta were logged in the same system or domain. If they would be logged in different systems, they would communicate via the gateway 68.

When an access point or a gateway receives a packet from the intranet, the access point or the gateway searches the fixed-length address from the memory by using the search means.

A packet to be transmitted in an intranet may include one, two, or more fixed-length addresses. If the packet includes at least two fixed-length addresses, one of the addresses may cause a predefined operation in a receiving access point or a gateway. The predefined operation may be, for example, the comparing of domain names as described above.

We may suppose that a fixed-length address is usually generated from one piece of information that identifies e.g. a sender, a receiver, or a service. However, a certain predefined combination of pieces of information could be one reference address and that reference address is inputted as a parameter to a hash function.

For example, a reference address could be composed of a name and a home address. Thus, a character string "Clay Saviranta, Luutnantintie 10 E 62, 00410 Helsinki, Finland" could be a reference address related to a certain person. The fixed-length address generated from this reference address could be a pointer to e.g. a health care database, so that the data collection of the said person is obtainable through the pointer. A fixed-length address generated from a name and a home address may be useful if a unique key, such as a social security number, is not available. A person's name or home address is alone an unreliable key, for example, to a health care database, because two or more persons may have the same name and members of a family usually have the same home address.

The combination of pieces of information may result in a longer reference address from which it is more reliable to generate a unique fixed-length address than one piece of information.

The combination of pieces of information can also be used to save the processing capacity of a server. This is possible, for example, in the following case. Let us suppose that 1) the server has access to connection information including sender and receiver information and 2) the sender and receiver information is reference addresses, both of them causing a search operation in the server. The sender information could be Alfa@wiral.com and the receiver information Beta@wiral.com. The server joins these character strings resulting in a reference address "Alfa@wiral.comBeta@wiral.com". Then the server generates a fixed-length address by applying an appropriate hash function to the said reference address. Hereafter the server creates a node including the fixed-length address and adds the node to the search means. During a connection, the sender sends messages to the receiver, wherein the messages contain the sender and receiver information. The server joins the sender and receiver information to one reference address, generates a fixed-length address for the said reference address, and uses the search means to find the fixed-length address. Thus, the server executes one search operation instead of two search operations.

The search means can be utilized in various client-server systems, such as an access point system show in FIG. 6.

FIG. 7A shows a client-server system in which a client 701 is a browser installed in a terminal 702 and a server 703 is software operating in a WWW server 704. The WWW server contains thousands of WWW pages stored in a database 705. The client 701 and the server 703 communicate via an access network 706. The client sends a URL/URI request concerning a WWW page to the server and the server sends the WWW page requested to the client.

FIG. 7B shows what happens in the server 703 when it receives the URL/URI request. The server is equipped with a search means including nodes 707 and 708 and, of course, a number of other nodes. Files 709 and 710 are stored in the database 705. The file 709 includes a reference address 712 from which a fixed-length address 713 is generated. The file 710 includes another reference address 714 from which another fixed-length

address 715 is generated. Let us suppose that the URL/URI request of the client includes the following reference address:

http://www.wiral.com/w_2/index.php?pgroup=products

The server 703 must find out whether the database includes a WWW page related to the reference address and if it does, send the WWW page to the client. The server generates a search address by applying a hash function to the reference address and uses the search means. In this case, the search address is found in the node 708 of the search means. The node 708 includes a pointer 716 pointing to the file that includes the reference address and the content of the WWW page requested.

Claims

1. A data structure for data collection searches from a set of data collections stored in a memory, a data collection of the set being related to at least one reference address, the data structure containing nodes each of which includes

a pointing means for accessing a certain data collection from the memory,

characterized in that each of said nodes further includes

a fixed-length address which is generated by applying a hash function to a reference address related to the certain data collection.

2. The data structure as defined in claim 1, characterized in that each of said nodes further includes at least one

pointing means adapted to link a node of the data structure to at least one other node of the data structure.

3. The data structure as defined in claim 1, characterized in that the data structure is a tree.

4. The data structure as defined in claim 3, characterized in that the data structure is a B-tree.

5. The data structure as defined in claim 1, characterized in that the data structure is a skiplist.

6. The data structure as defined in claim 1, characterized in that the hash function is essentially based on the MD5 message-Digest algorithm.

7. The data structure as defined in claim 1, characterized in that the reference address is a unique piece of information among reference addresses related to the set of data collections.

8. The data structure as defined in claim 1, characterized in that the reference address contains at least two pieces of information related to the certain data collection.

9. The data structure as defined in claim 1, characterized in that the reference address is of various length.

10. The data structure as defined in claim 1, characterized in that a data collection search is based on comparing a search address to the fixed-length address located in the node, wherein the search address is generated by applying a hash function to a search key.

11. A method for creating a data structure which is intended for searches from a set of data collections stored in a memory, a data collection of the set containing at least one reference address, each node of the data structure including

a pointing means for accessing a data collection from the memory,

characterized in that each of said nodes further includes a fixed-length address, the method containing the steps of:

receiving a data collection to be added to the set,

generating a fixed-length address by applying a hash function to a reference address obtained from the data collection,

locating the fixed-length address in a node of the data structure,

storing the data collection in the memory, and

setting a pointing means of the node to point to the data collection.

12. The method as defined in claim 11, characterized by the further steps of:

in a case where the node includes pointing means for linkage,

using said pointing means in accordance with a category of the data structure.

13. The method as defined in claim 11, characterized in that the category is a tree.

14. The method as defined in claim 13, characterized in that the category is a B-tree.

15. The method as defined in claim 11, characterized in that the category is a skiplist.

16. The method as defined in claim 11, characterized in that the hash function is essentially based on the MD5 message-Digest algorithm.

17. The method defined in claim 11, characterized in that the reference address is a unique piece of information among reference addresses related to the set of data collections.

18. The method as defined in claim 11, characterized in that the reference address contains at least two pieces of information related to the data collection.

19. A server for data collection searches applying to a set of data collections stored in the server, a data collection of the set being related to at least one reference address,

characterized in that the server includes at least one search means using records stored in memory each of which includes

a pointing means for accessing a certain data collection, and

a fixed-length address which is generated by applying a hash function to a reference address related to the certain data collection.

20. The server as defined in claim 19, characterized in that each of said records further includes at least one pointing means for linking the record to another record.

21. The server as defined in claim 19, characterized in that the records compose a tree data structure.

22. The server as defined in claim 21, characterized in that the records compose a B-tree.

23. The server as defined in claim 19, characterized in that the records compose a skiplist.

24. The server as defined in claim 19, characterized in that the hash function is essentially based on the MD5 message-Digest algorithm.

25. The server as defined in claim 19, characterized in that the reference address is a unique piece of information among reference addresses related to the set of data collections.

26. The server as defined in claim 19, characterized in that the reference address contains at least two pieces of information related to the data collection.

27. The server as defined in claim 19, characterized in that the reference address is a mobile subscriber integrated service digital network number (MSISDN).

28. The server as defined in claim 19, characterized in that the reference address is a fully qualified domain name (FQDN).

29. The server as defined in claim 19, characterized in that the reference address is a domain name.

30. The server as defined in claim 19, characterized in that the reference address contains a uniform resource locator (URL).

31. The server as defined in claim 19, characterized in that the server is coupled to at least one network.

32. The server as defined in claim 31, characterized in that the network is an access network.

33. The server as defined in claim 31, characterized in that the network is an intranet.

34. The server as defined in claim 31, characterized in that the server is further adapted to:

receive data from a client, said data containing at least one reference address,

generate a fixed-length address by applying a hash function to the reference address obtained from the data, and

perform a predetermined operation using the fixed-length address generated.

35. The server as defined in claim 34, characterized in that to perform the predetermined operation the server is adapted to:

search the fixed-length address from the records stored in the memory, and when found,

transmit the data to another client.

36. The server as defined in claim 34, characterized in that to perform the predetermined operation the server is adapted to:

search the fixed-length address from the records stored in the memory, and when found,

transmit response data to the client.

37. The server as defined in claim 34, characterized in that to perform the predetermined operation the server is adapted to:

locate at least the first fixed-length address in a packet containing the data and

send the packet to the intranet.

38. The server as defined in claim 33, characterized in that the server is adapted to:

receive a packet via the intranet, the packet containing a fixed-length address,

search the fixed-length address from the records stored in the memory, and when found,

perform a predetermined operation.

39. The server as defined in claim 38, characterized in that to perform the predetermined operation the server is adapted to:

search the fixed-length address from the records stored in the memory, and when found,

transmit the data to a certain client.

40. The server as defined in claim 38, characterized in that to perform the predetermined operation the server is adapted to:

search the fixed-length address from the records stored in the memory, and when found,

transmit the data to a certain gateway.

41. The server as defined in claim 35, characterized in that the server is a service access point.

42. The server as defined in claim 37 and 38, characterized in that the server is a service access point coupled with the intranet to at least one other server.

43. The server as defined in claim 40, characterized in that the server is a gateway.

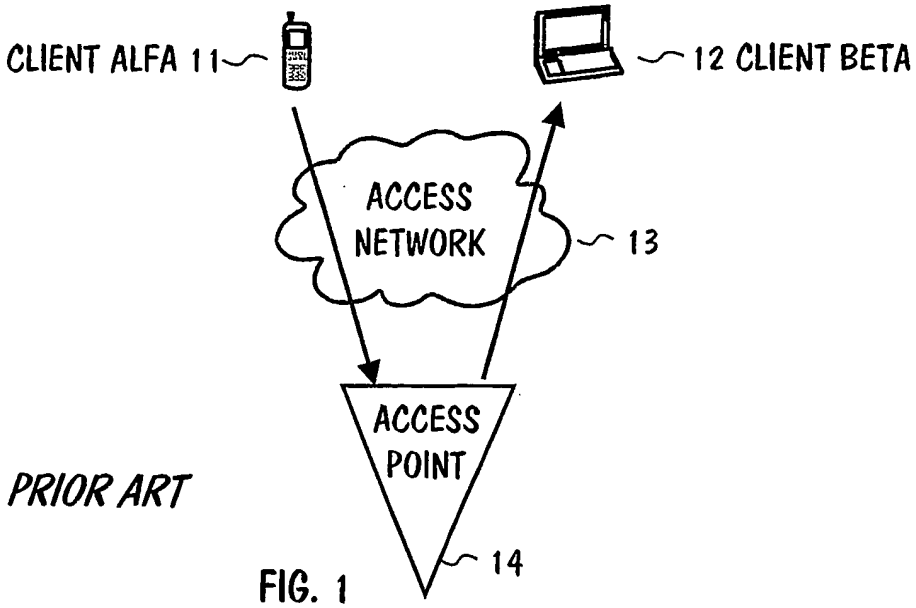
44. The server as defined in claim 34, characterized in that the data from the client is an instant message.

45. The server as defined in claim 36, characterized in that the server is a database server.

46. The server as defined in claim 34 and 45, characterized in that the data from the client is a database request.

47. The server as defined in claim 36, characterized in that the server is a WWW server.

48. The server as defined in claim 34 and 47, characterized in that the data from the client is a uniform resource locator (URL) request.



PRIOR ART

FIG. 1

PRIOR ART

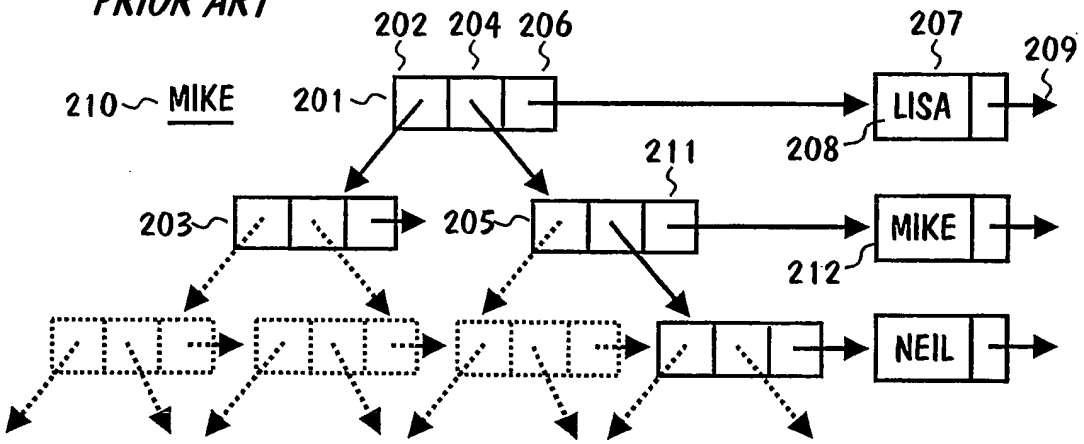


FIG. 2

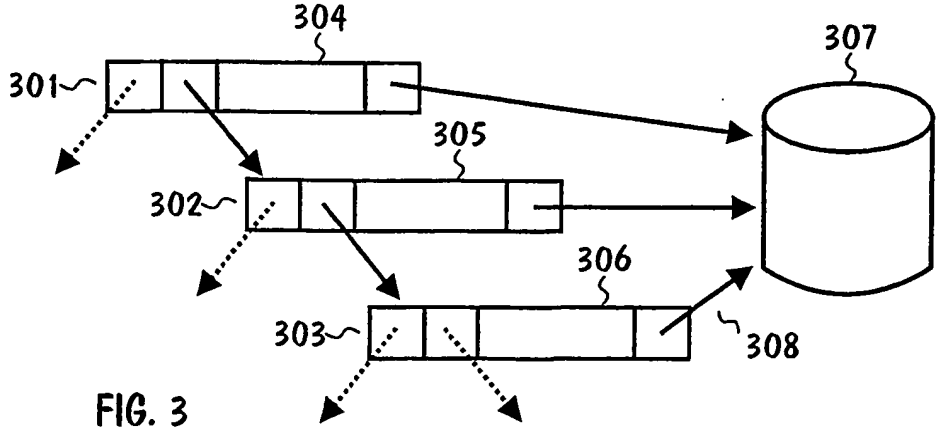


FIG. 3

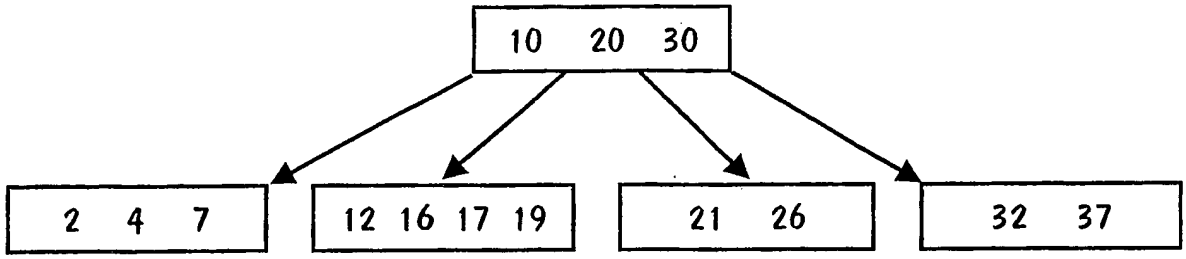


FIG. 4

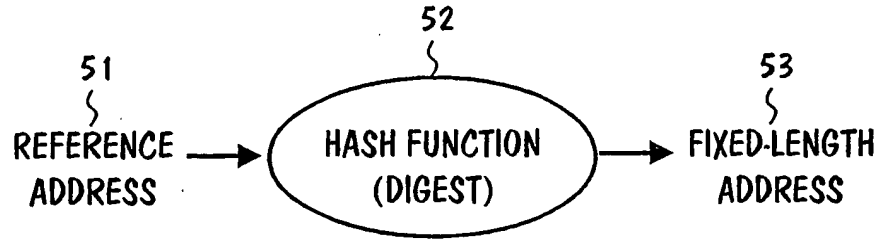


FIG. 5

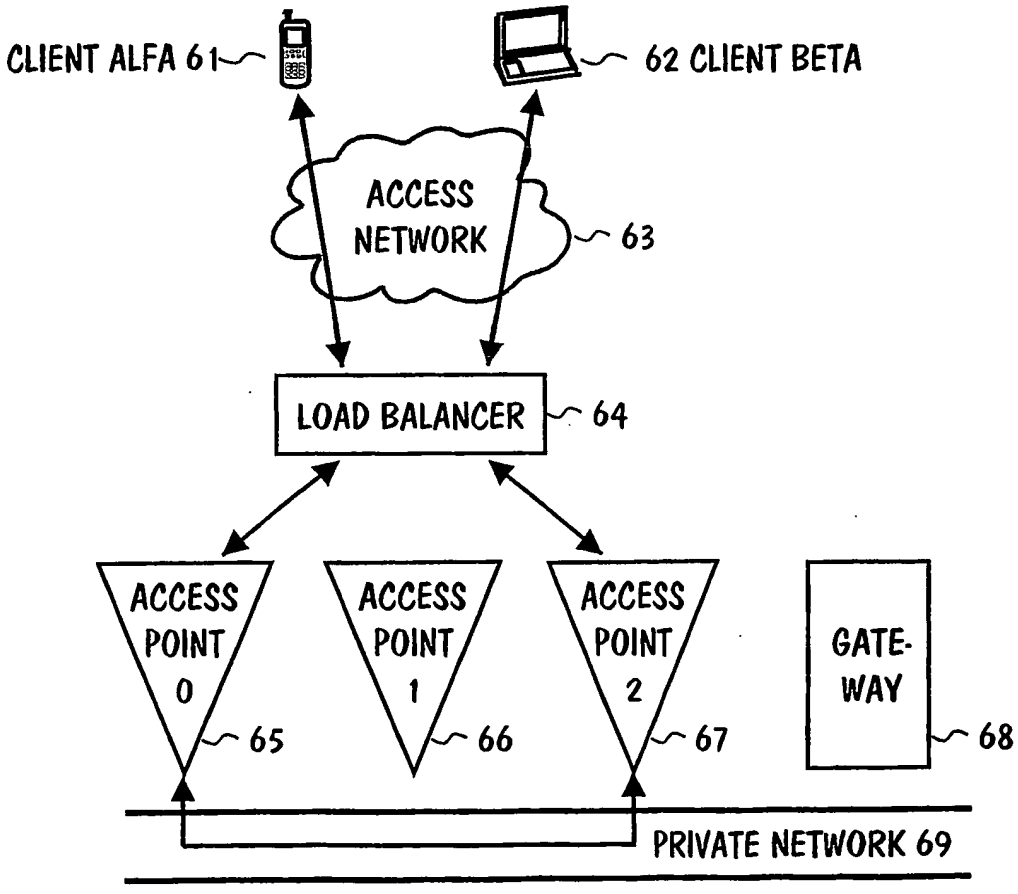


FIG. 6

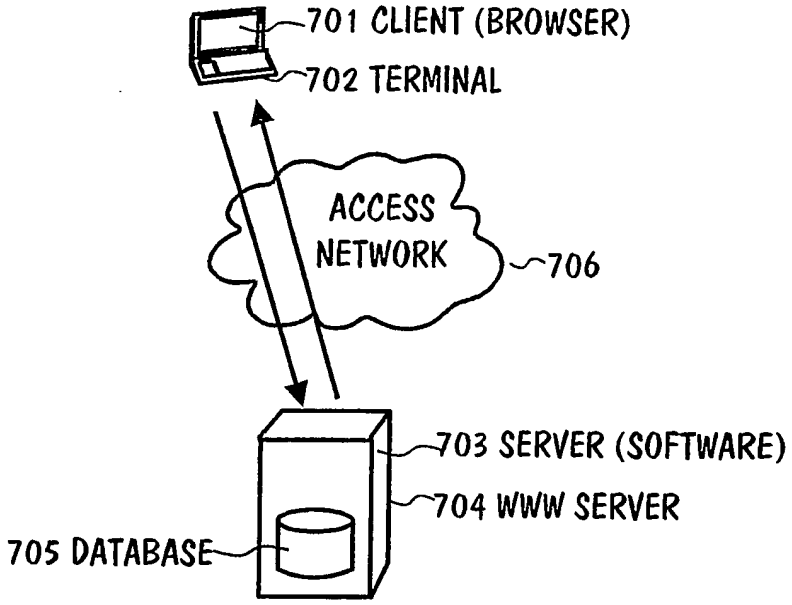

701 CLIENT (BROWSER)
702 TERMINAL
ACCESS NETWORK
706
703 SERVER (SOFTWARE)
704 WWW SERVER
705 DATABASE

FIG. 7A

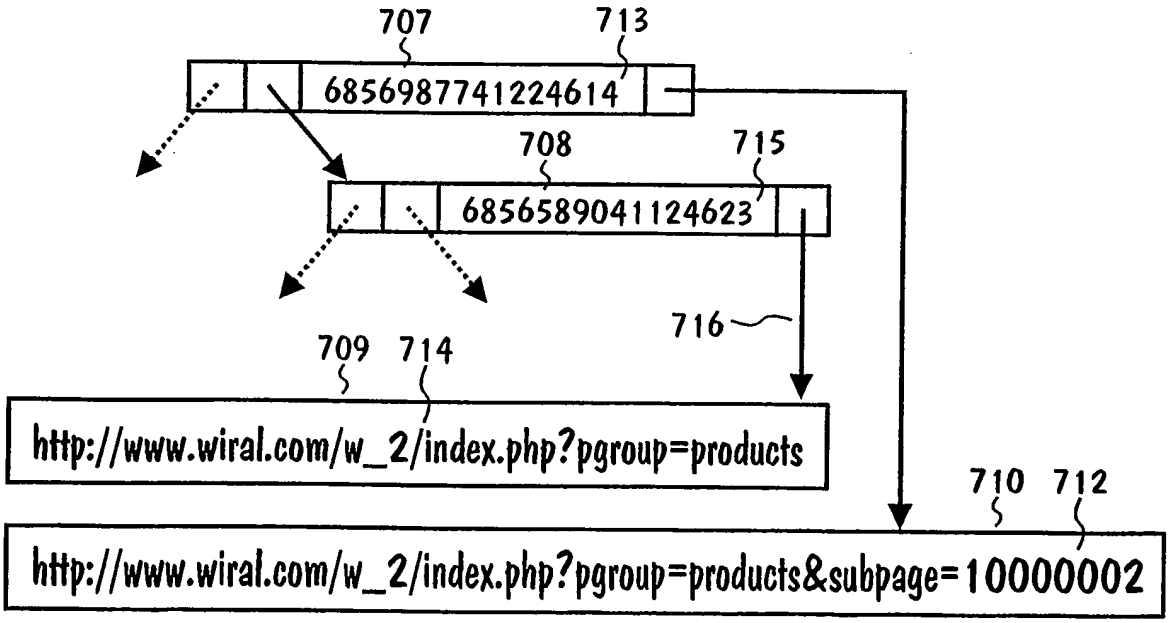

707
713
6856987741224614
708
715
6856589041124623
716
709
714
http://www.wiral.com/w_2/index.php?pgroup=products
710
712
http://www.wiral.com/w_2/index.php?pgroup=products&subpage=10000002

FIG. 7B

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 02/00257

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 17/30, H04L 29/12, H04L 12/56
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F, H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5914938 A (BRADY, D.M. ET AL.), 22 June 1999 (22.06.99), column 2, line 59 - column 3, line 27 --	1-48
X	US 5490258 A (FENNER, P.R.), 6 February 1996 (06.02.96), column 3, line 25 - line 38; column 3, line 57 - line 58; column 6, line 6 - line 17, column 18, line 11 - line 44 --	1-48
Y	WO 0021254 A2 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)), 13 April 2000 (13.04.00), page 13, line 8 - line 17 --	1-48

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 October 2002	18 -10- 2002
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM	Authorized officer Kristoffer Ogebjer/LR

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 02/00257

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 0133384 A1 (ALTA VISTA CO), 10 May 2001 (10.05.01), claim 1	1-48
	--	
Y	US 5940478 A (VAUDREUIL, G.M. ET AL.), 17 August 1999 (17.08.99), figures 16a-b, claims 1-15	1-48
	--	
Y	US 5692177 A (MILLER, J.W.), 25 November 1997 (25.11.97), claims 1,2	1-48
	--	
A	US 5892904 A (ATKINSON, R.G. ET AL.), 6 April 1999 (06.04.99), column 6, line 17 - line 27	1-48
	--	
A	EP 1096393 A2 (FUJITSU LTD), 2 May 2001 (02.05.01), claims 1-9	1-48
	--	
A	US 5414704 A (SPINNEY, B.A.), 9 May 1995 (09.05.95), claim 1	1-48
	--	

INTERNATIONAL SEARCH REPORT

Information on patent family members

30/09/02

International application No.
PCT/FI 02/00257

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5914938 A	22/06/99	NONE	
US 5490258 A	06/02/96	US 5860136 A	12/01/99
		US 5842224 A	24/11/98
WO 0021254 A2	13/04/00	AU 1301600 A	26/04/00
		EP 1119986 A	01/08/01
WO 0133384 A1	10/05/01	US 6301614 B	09/10/01
US 5940478 A	17/08/99	AU 719138 B	04/05/00
		AU 3310197 A	05/01/98
		CA 2255933 A	04/12/97
		EP 0897632 A	24/02/99
		JP 2000512096 T	12/09/00
		KR 2000016304 A	25/03/00
		WO 9745991 A	04/12/97
US 5692177 A	25/11/97	NONE	
US 5892904 A	06/04/99	NONE	
EP 1096393 A2	02/05/01	JP 2001127794 A	11/05/01
US 5414704 A	09/05/95	DE 69324204 D,T	23/12/99
		EP 0594196 A,B	27/04/94